Ministry of higher education and scientific research

Diyala University

College of engineering

Design And Implementation for Line Follower of Mobile Robot

A project Submitted to Department of Electronic Engineering - College of Engineering - Diyala University in Partial fulfillment of the Requirements for the Degree of B. Sc in Electronic Engineering.

By :

Bashaar Abdul-Salam

Shahad Mohammed

Supervisor

Dr. Mohammad . S.Salh

٢٠١٥ – ٢٠١٦

وزارة التعليم العالي والبحث العلمي

جامعة ديالى كلية الهندسة

تصميم وبناء انسان الي جوال متتبع للخط

هذا المشروع قدم الى قسم الالكترونيك - كلية الهندسة – جامعة ديالى كجزء من متطلبات نيل شهادة البكلوريوس علوم الهندسة

اعداد الطالبات :

بشائر عبد السلام شاكر

شهد محمد عبد الكريم

باشراف

م. د. محمد سلمان صالح

2015 - 2016

Chapter One

INTRODUCTION

Chapter two

MOBILE ROBOT

Chapter three

ARDUINO

Chapter four

IMPLEMENTED SYATEM

Chapter five

CONCLUSION AND FUTURE WORK

Chapter One

INTRODUCTION

1.1 Introduction:

In the early 1800's mechanical puppets were first built in Europe, just for entertainment value. And these were called robots since their parts were driven by linkage and cams and controlled by rotating drum selectors. In 1801 Joseph Maria Jacquard made the next great change and invented the automatic draw loom. The draw looms would punch cards and was used to control the lifting of thread in fabric factories. This was the first to be able to store a program and control a machine. After that there were many small changes in robotics. The first industrial robots were Unimates developed by George Devol and Joe Engelberger in the late 50's and early 60's. The first patent was by Devol but Engelberger formed Unimation which was the first market robots. So Engelberger has been called the "father of robotics". For a while the economic viability of these robots proved disastrous and thing slowed down for robotics. But the industry recovered and by the mid-80's robotics was back on track.[1]

George DevolJr, in 1954 developed the multi jointed artificial arms which lead to the modern robots. But mechanical engineer Victor Scheinman developed the truly flexible arm known as the Programmable Universal Manipulation Arm (PUMA) [1]. In 1950 Isaac Asimov came up with laws for robots and these were:

i. A robot may not injure a human being, or through inaction allow a human being to come to harm.

ii. A robot must obey the orders given it by human beings, except where such orders would conflict with the first law

iii. A robot must protect its own existence as long as such protection does not conflict with the first or second law.[2]

Mobile Robotics moved into its own in 1983 when Odetics introduced this six-legged vehicle which was capable of climbing over objects. This robot could lift over 5.6 times its own weight parked and 2.3 times it weight moving. [3] In 2000 Sony unveils humanoid robots, the Sony Dream Robots (SDR) at Robodex. SDR is able to recognize 10 different faces, expresses emotion through speech and body language, and can walk on flat as well as irregular surfaces. In 2005 the Korean Institute of Science and Technology (KIST), creates HUBO, and claims it is the smartest robot in the world. This robot is linked to a computer via a high speed wireless connection; the computer does all of the thinking for the robot.[1]

1.2 Application area:

Line followers can be used to deliver mail within an office building and deliver medications in a hospital. The technology has been suggested for running buses and other mass transit systems, and may end up as part of autonomous cars navigating the freeway. The line follower can be used in guidance system for industrial robots moving on shop floor. An example might be in a warehouse where the robots follow 'tracks' to and from the shelves they stock and retrieve from. A line follower robot can be used in military as spy kids or in many other applications.[4]

Chapter two

MOBILE ROBOT

2.1 Introduction:

Robotics has achieved its greatest success to date in the world of industrial manufacturing. Robot arms, or *manipulators*, comprise a 2 billion dollar industry. Bolted at its shoulder to a specific position in the assembly line, the robot arm can move with great speed and accuracy to perform repetitive tasks such as spot welding and painting (figure 2.1). In the electronics industry, manipulators place surface-mounted components with superhuman precision, making the portable telephone and laptop computer possible. Yet, for all of their successes, these commercial robots suffer from a fundamental disadvantage: lack of mobility. A fixed manipulator has a limited range of motion that depends Picture of auto assembly plant-spot welding robot of KUKA and a parallel robot Delta of SIGDemaurexSA (invented at EPFL [140]) during packaging of chocolates.on where it is bolted down. In contrast, a mobile robot would be able to travel through outthe manufacturing plant, flexibly applying its talents wherever it is most effective. This book focuses on the technology of mobility: how can a mobile robot move unsupervised through real-world environments to fulfill its tasks? The first challenge is locomotion itself. How should a mobile robot move, and what is it about a particular locomotion mechanism that makes it superior to alternative locomotion mechanisms? Hostile environments such as Mars trigger even more unusual locomotion mechanisms (figure 2.2). In dangerous and inhospitable environments, even on Earth, such *teleoperated*systems have gained popularity. In these cases, the low-level complexities of the robot often make it impossible for a human operator to directly control its motions. The human performs localization and cognition activities, but relies on the robot's control scheme to provide motion control. For example, Plustech's walking robot provides automatic leg coordination

while the human operator chooses an overall direction of travel (figure 2.1).depicts an underwater vehicle that controls six propellers to autonomously stabilize the robot submarine in spite of underwater turbulence and water currents while the operator chooses position goals for the submarine to achieve.

Other commercial robots operate not where humans *cannot* go but rather share space with humans in human environments. These robots are compelling not for reasons of mobility but because of their *autonomy*, and so their ability to maintain a sense of position and to navigate without human intervention is paramount [5]

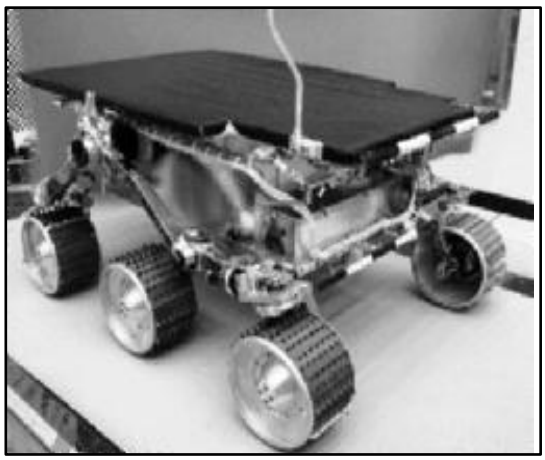


Figure (2.1): legged robot

The mobile robot Sojourner was used during the Pathfinder mission to explore Mars in summer 1997. It was almost completely teleoperated from Earth. However, some on-board sensors allowed for obstacle detection.(http://ranier.oact.hq.nasa.gov/telerobotics_page/telerobotics.shm). © NASA/JPL[5]

Mobile robotics is a young field. Its roots include many engineering and science disciplines,from mechanical, electrical and electronics engineering

to computer, cognitive and social sciences. Each of these parent fields has its share of introductory textbooks that excite and inform prospective students, preparing them for future advanced coursework and research. Our objective in writing this textbook is to provide mobile robotics with such a preparatory guide.This book presents an introduction to the fundamentals of mobile robotics, spanning the mechanical, motor, sensory, perceptual and cognitive layers that comprise our field of study. A collection of workshop proceedings and journal publications could present the new student with a snapshot of the state of the art in all aspects of mobile robotics. But here we aim to present a foundation a formal introduction to the field. The formalism and analysis herein will prove useful even as the frontier of the state of the art advances due to the rapid progress in all of mobile robotics' sub-disciplines.We hope that this book will empower both the undergraduate and graduate robotics student with the background knowledge and analytical tools they will need to evaluate andThe origins of the this book bridge the Atlantic Ocean. Theauthors have taught courses on Mobile Robotics at the undergraduate and graduate level at Stanford University, ETH Zurich, Carnegie Mellon University and EPFL (Lausanne). Their combined set of curriculum details and lecture notes formed the earliest versions of this text. We have combined our individual notes, provided overall structure and then test-taught using this textbook for two additional years before settling on the current, published text. For an overview of the organization of the book and summaries of individual chapters.[6]

Finally, for the teacher and the student: we hope that this textbook proves to be a fruitful launching point for many careers in mobile robotics. That would be the ultimate reward.even critique mobile robot proposals and artifacts throughout their career. This textbook is suitable as a whole for

introductory mobile robotics coursework at both the undergraduate and graduate level. Individual chapters such as those on Perception or Kinematics can be useful as overviews in more focused courses on specific sub-fields of robotics.[6]

2.2: Legged Mobile Robots:

Legged locomotion is characterized by a series of point contacts between the robot and the ground. The key advantages include adaptability and maneuverability in rough terrain. Because only a set of point contacts is required, the quality of the ground between those points does not matter so long as the robot can maintain adequate ground clearance. In addition, a walking robot is capable of crossing a hole or chasm so long as its reach exceeds the width of the hole. A final advantage of legged locomotion is the potential to manipulate objects in the environment with great skill. An excellent insect example, the dung beetle, is capable of rolling a ball while locomoting by way of its dexterous front legs. The main disadvantages of legged locomotion include power and mechanical complexity. The leg, which may include several degrees of freedom, must be capable of sustaining part of the robot's total weight, and in many robots must be capable of lifting and lowering the robot. Additionally, high maneuverability will only be achieved if the legs have a sufficient number of degrees of freedom to impart forces in a number of different directions.[6]

2.3: Analysis of Line Sensor Configuration for the Advanced Line Follower Robot:

A line follower robot is basically a robot designed to follow a 'line' or path already predetermined by the user. This line or path may be as simple as a physical white line on the floor or as complex path marking schemes

e.g. embedded lines, magnetic markers and laser guide markers. In order to detect these specific markers or 'lines', various sensing schemes can be employed [1]. These schemes may vary from simple low cost line sensing circuit to expansive vision systems. The choice of these schemes would be dependent upon the sensing accuracy and flexibility required. From the industrial point of view, line following robot has been implemented in semi to fully autonomous plants. In this environment, these robots functions as materials carrier to deliver products from one manufacturing point to another where rail, conveyor and gantry solutions are not possible[1]. Apart from line following capabilities, these robots should also have the capability to navigate junctions and decide on which junction to turn and which junction ignore. This would require the robot to have 90 degree turn and also junction counting capabilities. To add on to the complexity of the problem, sensor positioning also plays a role in optimizing the robots performance for the tasks mentioned earlier[2]. This paper attempts to present a simple set of experiments on sensor positioning and also controlling strategy to enable junction counting and also 90 degree turn accuracy. These strategies are tested on a basic test robot base (refer to figure 1) system on a test pitch based upon ROBOCON 2006 [7]requirement as shown in figure(2.2)

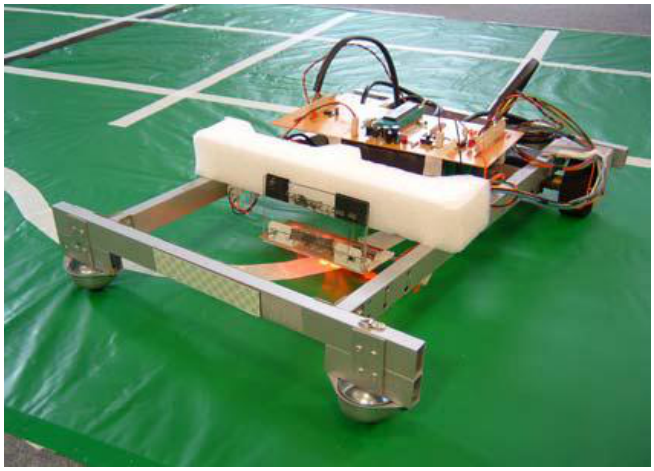


Figure (2.2): Advanced Line Follower (ALF) Robot

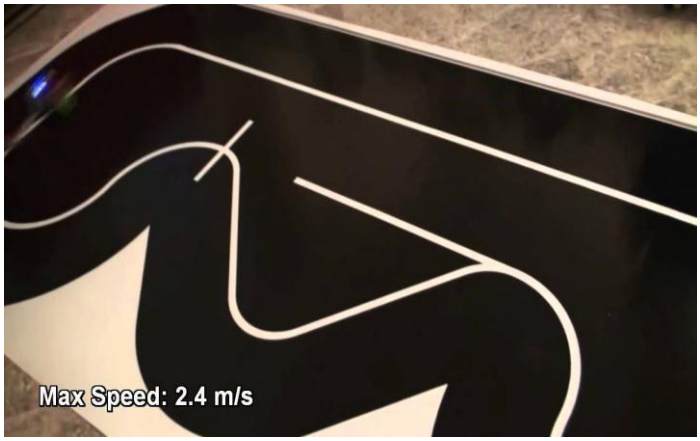


Figure (2.3): ROBOCON 2006 pitch requirement

2.4: Application:

• The robots have potential application in areas where a vehicle or a mechanic automatic system may exist

- Areas of application:

1. Support to medical services – SERVICE ROBOTS
a. Transportation of food, medication, medical exams
b. Automation of pharmacy service
2. Automatic cleaning of (large) area
a. Supermarkets, airports, industrial sites
b. Glass cleaning
c. Domestic vacuum-cleaner
3. Client support
a. Museum tours, exhibitions guides
4. Agricultural
a. Fruit and vegetable picking, fertilization, planting
5. Forests
a. Cleaning, fire preventing, tree cutting

b. Areas of application (ctn):

6. Hazard Environments

a. Inspection of hazard environments (catastrofic areas, vulcanos, nuclear

7. power plants, oil tanqs)

a. Inspection of gas or oil pipes, and power transmis

b. Oil tank cleaning

8. Construction and demolishing

9. Space

a. Space exploration

b. Remote inspection of space stations

10. Military

a. Surveilance vehicles

b. Monitoring vehicles

11. Forests

a. Cleaning, fire preventing, tree cutting

b. Areas of application (ctn):

12. Material Handling

a. AGVs, SGVs, LGVs

13. Safety

a. Surveillance of large areas, buildings, airports, car parking lots

14. Civil Transportation

a. Inspection of airplanes, trains,

15. Elderly and Handicapped

a. Assistance to handicapped or elderly people, helping in tansportation, health care,

16. Entertainment

a. RobotDog

b. Aibo – Robot dog from Sony [7]

Chapter three

ARDUINO

3.1 Introduction:

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source,which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time.For advanced Arduino users, prowl the web; there are lots of resources.The Arduino project was started in Italy to develop low cost hardware for interaction design. An overview is on the Wikipedia entry for Arduino. The Arduino home page is http://www.arduino.cc/.The Arduino hardware comes in several flavors. In the United States, Sparkfun (www.sparkfun.com) is a good source for Arduino hardware.This guide covers the Arduino Uno board (Sparkfun DEV-09950, $29.95), a good choice for students and educators. With the Arduino board, you can write programs and create interface

circuits to read switches and other sensors, and to control motors and lights with very little effort. Many of the pictures and drawings in this guide were taken from the documentation on the Arduino site, the place to turn if you need more information. The Arduino section on the ME 2011 web site, https://sites.google.com/a/umn.edu/me2011/, covers more on interfacing the Arduino to the real world..[8] As shown in Fig(3.1): below

Fig (3.1): Ardunio UNO

The Arduino programming language is a simplified version of C/C++. If you know C,programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions. An important feature of the Arduino is that you can create a control program on the host PC,download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.[8]

Arduino is an open source physical computing platform based on a simple input/output (I/O) board and a development environment that implements the Processing language (www.processing.org). Arduino can be used to develop standalone interactive objects or can be connected to software on your computer (such as Flash, Processing, or Max/MSP). The boards can be assembled by hand or purchased preassembled; the open source IDE (Integrated Development Environment) can be downloaded for free from www.arduino.cc Arduino is different from other platforms on the market because of these

features:

1. It is a multiplatform environment; it can run on Windows, Macintosh,and Linux
2. It is based on the Processing programming IDE, an easy-to-use development environment used by artists and designers

3. You program it via a USB cable, not a serial port. This feature is useful,because many modern computers don't have serial ports.

4. It is open source hardware and software—if you wish, you can download the circuit diagram, buy all the components, and make your own, without paying anything to the makers of Arduino

5. The hardware is cheap. The USB board costs about €20 (currently, A bout US$35) and replacing a burnt-out chip on the board is easy and costs no more than €5 or US$4. So you can afford to make mistakes.

6. There is an active community of users, so there are plenty of people who can help you.

7. The Arduino Project was developed in an educational environment and is therefore great for newcomers to get things working quickly.

3.2 What Is Physical Computing?

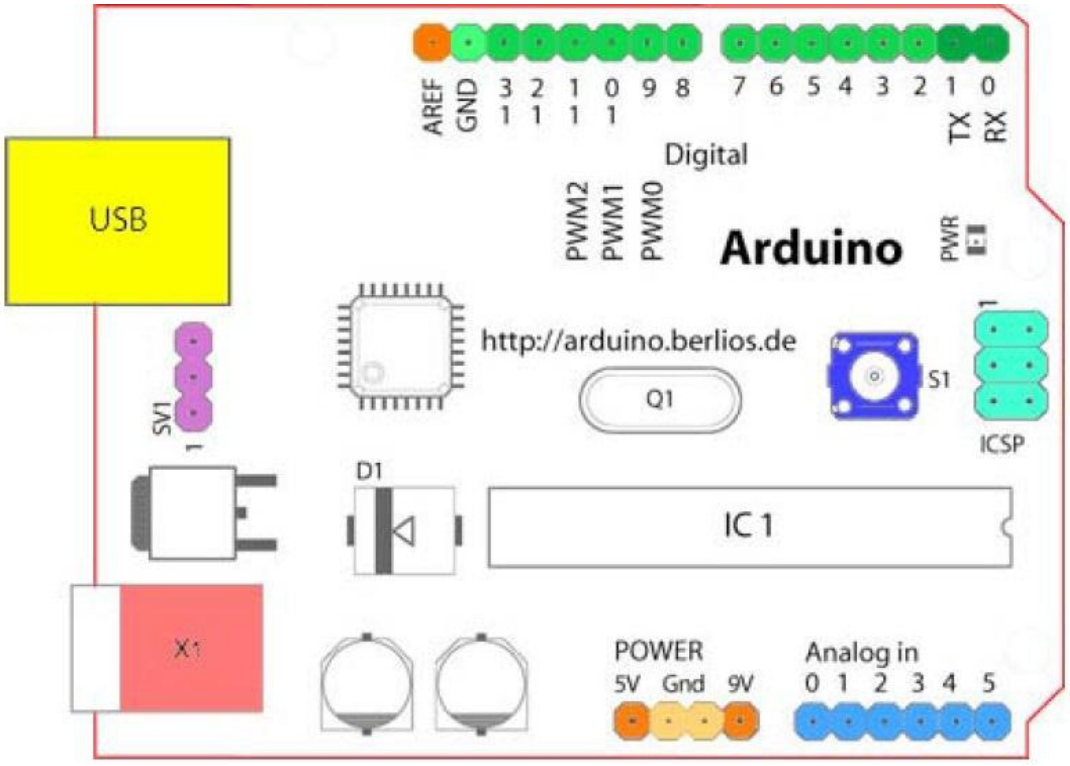
Physical Computing uses electronics to prototype new materials for designers and artists. It involves the design of interactive objects that can communicate with humans using sensors and actuators controlled by a behaviour implemented as software running inside a microcontroller (a small computer on a single chip). In the past, using electronics meant having to deal with engineers all the time, and building circuits one small component at the time; these issues kept creative people from playing around with the medium directly. Most of the tools were meant for engineers and required extensive knowledge. In recent years, microcontrollers have become cheaper and easier to use, allowing the creation of better tools.The progress that we have made with Arduino is to bring these tools one step closer to the novice, allowing people to start building stuff after only two or three days of a workshop. With Arduino, a designer or artist can get to know the basics of electronics and sensors very quickly and can start building prototypes with very little investment.

The Arduino philosophy is based on making designs rather than talking about them. It is a constant search for faster and more powerful ways to build better prototypes. We have explored many prototyping techniques and developed ways of thinking with our hands. Classic engineering relies on a strict process for getting from A to B; the Arduino Way delights in the possibility of getting lost on the way and finding C instead.This is the tinkering process that we are so fond of—playing with the medium in an open-ended way and finding the unexpected. In this search for ways to build better prototypes, we also selected a number of software packages that enable the process of constant manipulation of the software and hardware medium. The next few sections present some philosophies, events, and pioneers that have inspired the Arduino Way.

3.3 Arduino UNO:

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, aUSB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards. The Arduino Uno can be

powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.[9]



The pins of the arduino UNO are shown in fig (3.2)

Power pins are as follows:

1. Th VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
2. 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
3. 3V3. A 3.3 volt supply generated by the on-board regulator Maximum current draw is 50 mA.
4. GND. Ground pins.

3.3.1 Input And Output

Each of the 14 digital pins on the Uno can be used as input or output, using pinMode(), digitalWrite(), anddigitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions: Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip. External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details. PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off [9]

3.3.2 Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required.. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-toserial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.[9]

3.3.3 Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials. The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for

details. The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).[9]

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to

re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.[9]

3.3.4 USB Overcurrent Protection

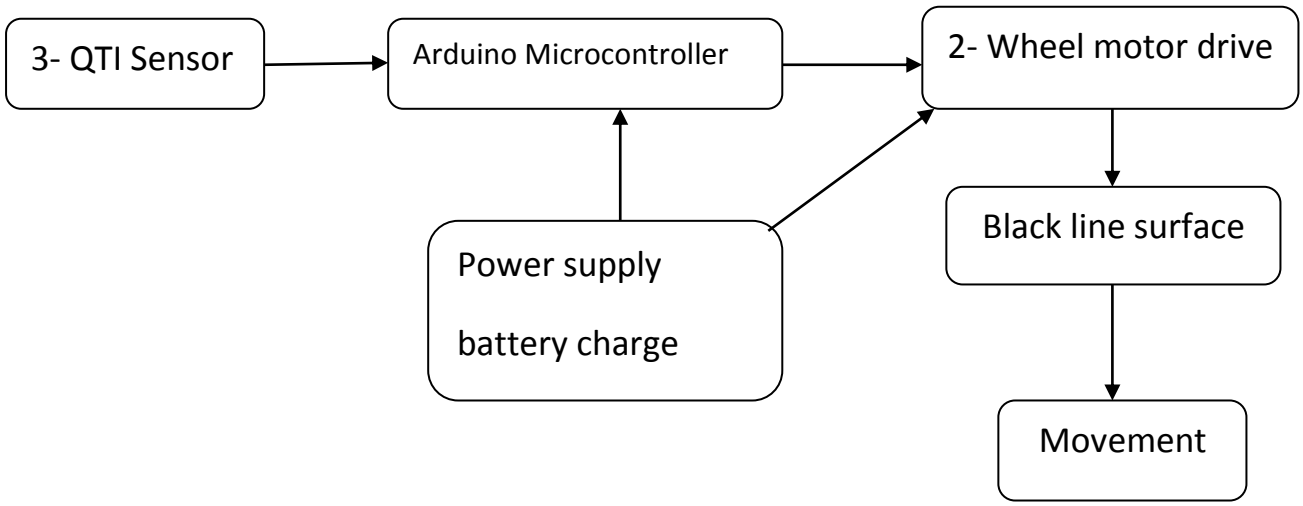
The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.[6] The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multipleof the 100 mil spacing of the other pins.[9]

Chapter four

IMPLEMENTED SYATEM

4.1 Introduction:

In this project the two wheel forward motor used to determine the direction with one at the back is used to assign only the following components were used in the project we used the microcontroller ardunio device and the use of three sensor from type QTI sensor and the number of QTI are used is 3 first of the QTI sensor connector to the left and the second of the QTI sensor connector to the center and the third connector to the right and also used dc motor drives .AS to explain at this paper and shown the block diagram of line follower mobile robot:



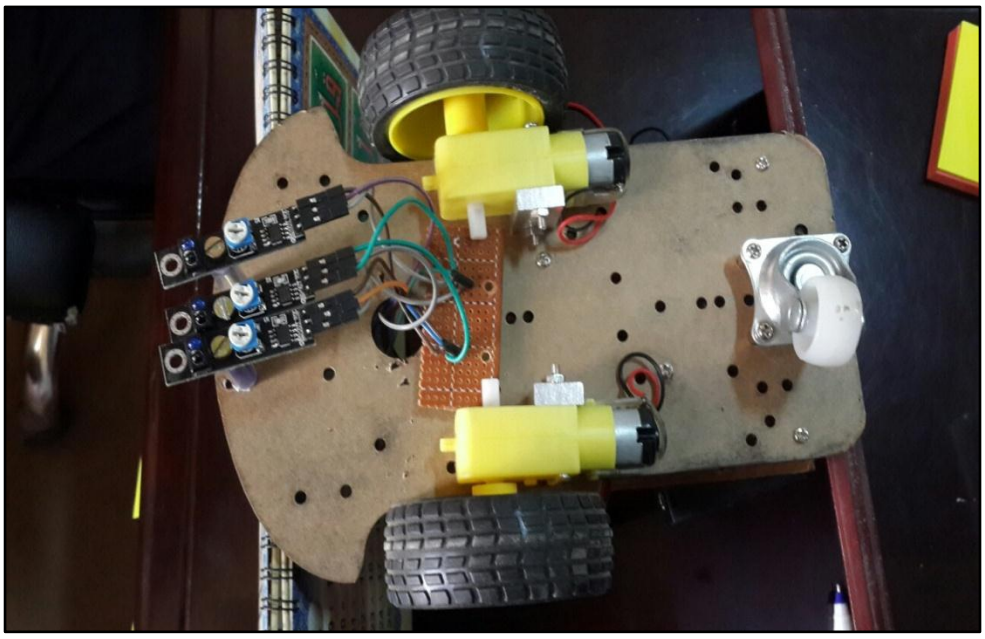
Fig(4.1):Block diagram of line follower

4.2 Motors & Wheels:

4.2.1 Driven Wheels:

Weight-bearing wheels not driven by motors, directly or indirectly, are considered driven wheels. Driven wheels unable to swivel left and right can cause excess friction when the robot turns. Caster wheels are a special kind of driven wheel that allows the wheels to swivel left and right,

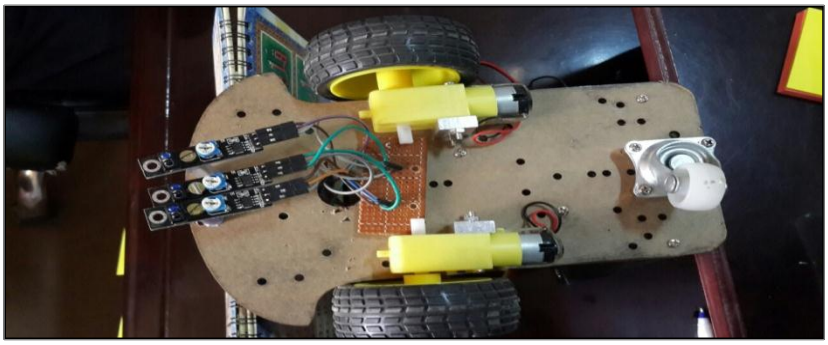
eliminating unnecessary friction when the robot turns.[10] As shown in fig below:



Fig(4.2): Driven wheel

4.2.3 Omni-Directional Wheels:

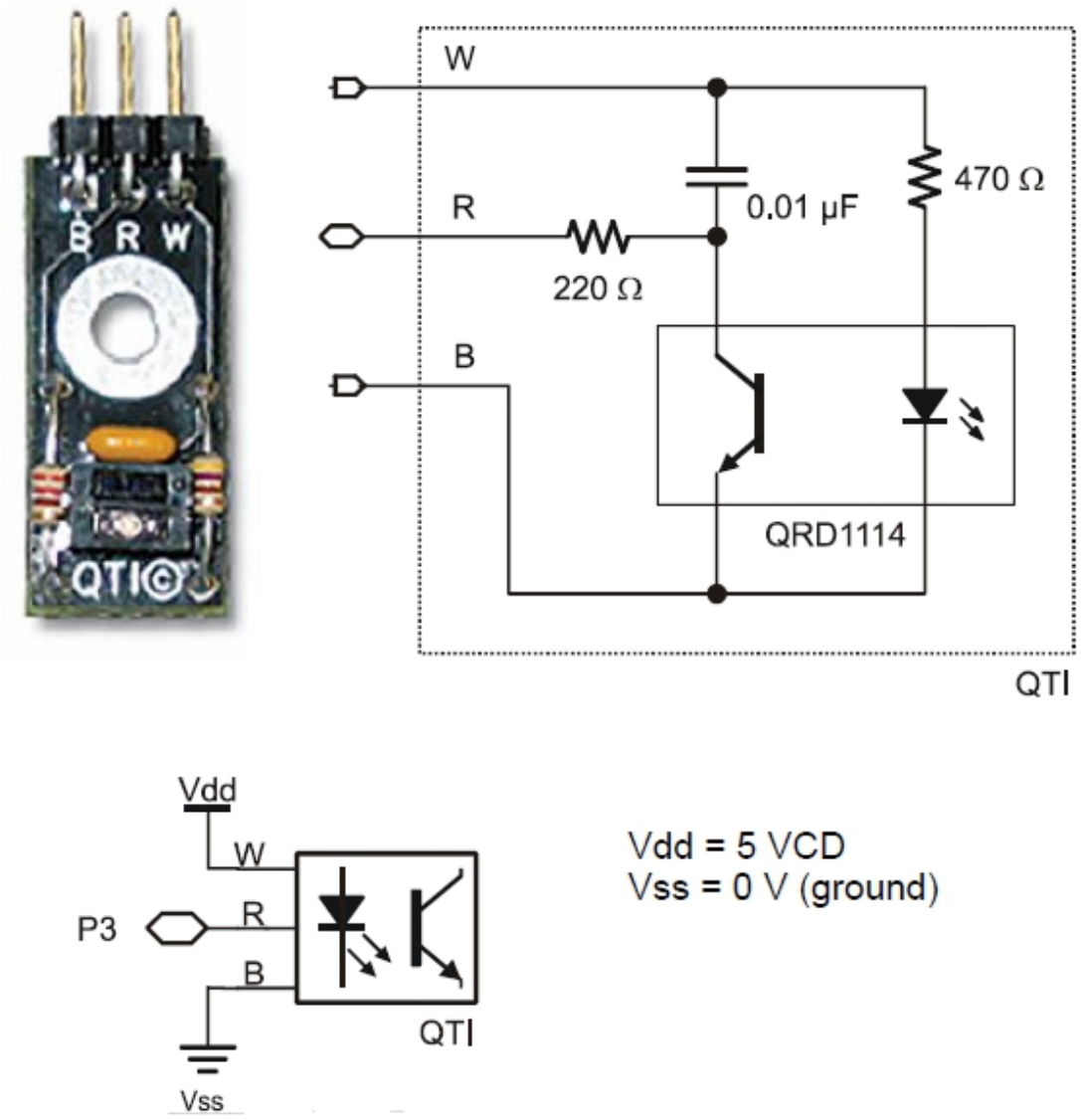
The TETRIX Omni-directional wheels, or Omni-wheels, are comprised of one large wheel with many smaller wheels along its circumference. The smaller wheels are perpendicular to the main wheel, allowing it to roll both forward and backward, and side-to-side, naturally.[10]



Fig(4.3):Wheel backing

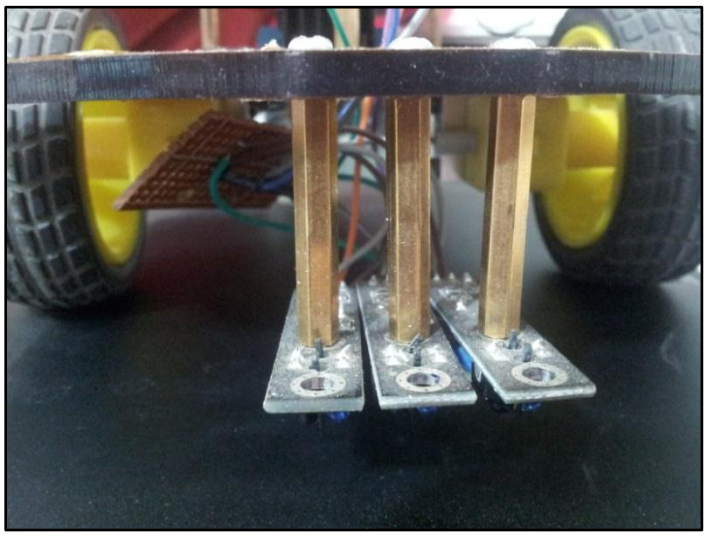
4.3 QTI Sensor :

The QTI sensor is a close-proximity infrared emitter and receiver pair mounted on a tiny PCB. It can be used as an analog sensor to differentiate between different levels of infrared reflectivity. It can also be used as a purely digital device that returns a 1 when it detects a black line or a 0 if it detects a white background. An array of four QTI sensors used as digital devices makes an effective and flexible line-follower for your small robot.**[11]**



Fig(4.4): QTI sensor

It was used in this project 3 QTI Sensor uses a first of sensor for sensing motor wheel drive on the right and the second QTI is used to sensing motor wheel drive on the left and the third QTI Sensor used to sensing for the middle. As shown in figure (4:5) below:

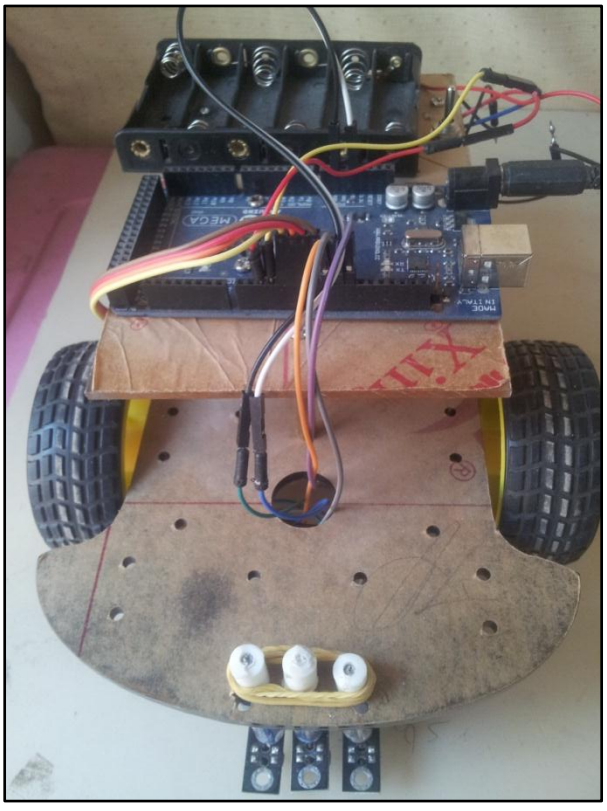


Fig(4.5): QTI Sensor

4.4 Arduino UNO:

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, aUSB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one

in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards. The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.[9] As shown in fig.(4.6).



Fig(4.6): Ardunio Microcontroller

There are two line test the robot to follow there as shown in fig.(4:7).

Fig.(4.7): Two Line track

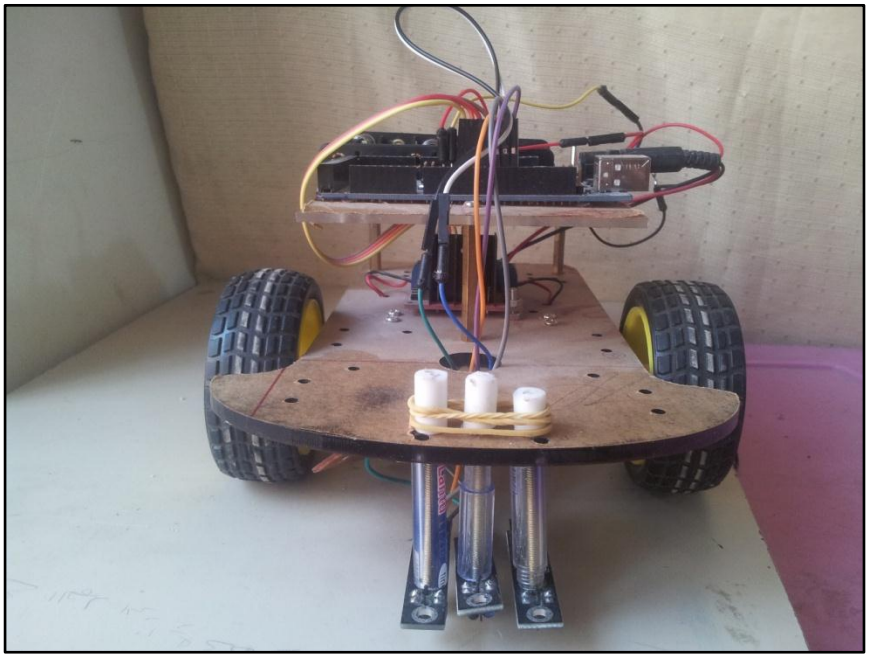
4.5 Software

The line used in the project is Black color and used tipp for the path Selection and it was select many paths to be tested robot.

4.7 movement:

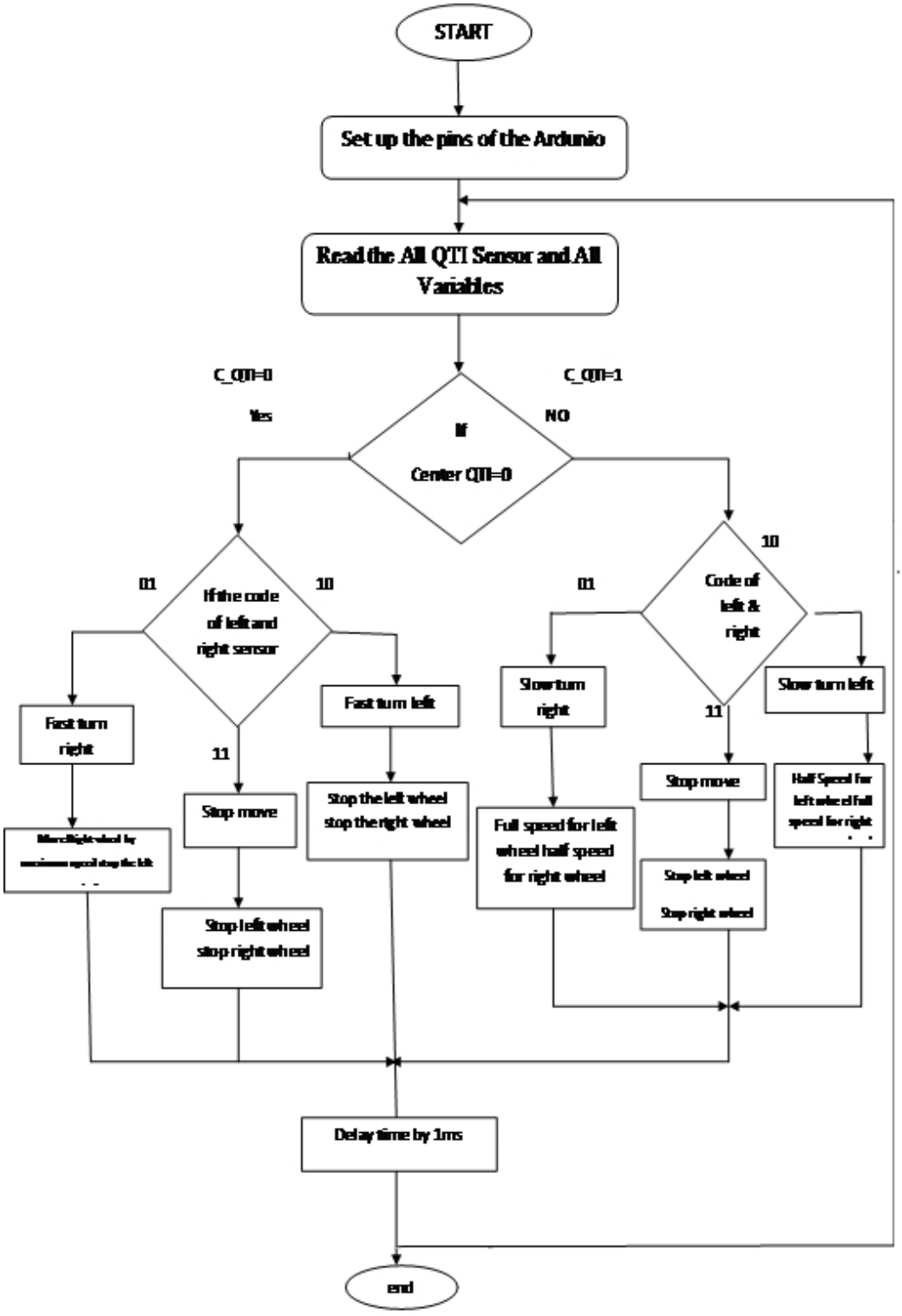
Motion shall be according to the path that determines.

And that the figure of this project:



Fig(4.8): line follower mobile robot

The microcontroller is read the output of the QTI sensoers then make the dicession of the movement forward, turn left or right fast or slow or stop when all the sensor detect black serface (output signal of QTI sensoris 1). The flowchart of the program is shown in fog.(4.9):



Fig(4.9): flow chart of the line follower mobile robot

And the code is:

```
//Line follower_Robort
int RS=8;
int CS=9;
int LS =10;
int spd _LM=3;
int spd_RM=2;

int LM_A=4;
int LM_B=5;
int RM_A=6;
int RM_B=7;
int spd_100;
int backmove_index=0;

int rv,cv,lv;
voidsetup()
{
pinMode(RS,INPUT);
pinMode(CS,INPUT);
pinMode(LS,INPUT);

PINMode(spd_LM,OUTPUT);
PINMode(spd_RM,OUTPUT);
pinMode(LM_A,OUTPUT);
pinMode(LM_B,OUTPUT);

pinMode(RM_A,OUTPUT);
```

```
pinMode(RM_B,OUTPUT);

digitalWrite(LM_A,HIGH);
digitalWrite(LM_B,LOW);

digitalWrite(RM_A,HIGH);
digitalWrite(RM_B,LOW);

analogWrite(spd_LM,0);

analogWrite(spd_RM,0);

delay(1000);
serial.begin(9600);
}
Void loop()
{
rv=digitalRead(RS);
cv=digitalRead(CS);
lv=digitalRead(LS);
Searial.print(rv);
Searial.print('\t');
Searial.print(cv);
Searial.print('\t');
Searialprintin(iv);

// //delay(100);
If(backmove_index==1)
{if(rv==1 || cv==1 || lv==1)
```

```
{correct_move();}
}
If(cv==1)// center at black line
{
If(rv==0 && lv==0)    forward();
else if(rv==1 && lv==0)  slow_left();
else if(rv==0 && lv==1)  slow_right();
else if(rv==1 && lv==1)  stop_move ();
}
else //if(cv==0)center at waith back graound
}
If(rv==1 && lv==1)   stop_move();
else If(rv==1 && lv==0)   fast_left();
else If(rv==0 && lv==1)   fast_right();
// else If(rv==0 && lv==0)   back_move();//search();
}
//delay(10);
}
////////////
Void forward()
{
analogWrite(spd_LM,spd);
analogWrite(spd_RM,spd);
}
Void fast_left()
{
AnalogWrite(spd_LM,0);
AnalogWrite(spd_RM,spd);
}
```

```
void slow_left()
{
analogWrite(spd_LM,spd/2);
analogWrite(spd_RM,spd);
}
void fast_right()
{
analogWrite(spd_LM,spd);
analogWrite(spd_RM,0);
}
void slow_right()
{
 analogWrite(spd_LM,spd);
analogWrite(spd_RM,spd/2);
}
void stop_move()
}
analogWrite(spd_LM,0);
analogWrite(spd_RM,0);
}
Voidback_move(void)
{
// if (backmove_index==1)
//{
Stop_move();
digitalWrite(LM_A;LOW);
digitalWrite(LM_B;HIGH);

digitalWrite(RM_A;LOW);
```

```
digitalWrite(RM_B;HIGH);
forward();
backmove_index=1;
//}

}
void correct-move(void)
{
Stop_move();
digitalWrite(LM_A;HIGH);
digitalWrite(LM_B;LOW);

digitalWrite(RM_A;HIGH);
digitalWrite(RM_A;LOW);
backmove_index=0;
}
```

Chapter five

CONCLUSION AND

5.1 Conclusion:

The Line follower robot works successfully to track on the black line. Above the white surface (art paper) there are some black lines in different directions. The robot still good enough to sense the line and follows the track**.**

5.2 Future work:

1- Software control of the line type (dark or light) to make automatic detection possible.
2- “Obstacle detecting sensors” to avoid physical obstacles and continue on the line.
3- Distance sensing and position logging & transmission

1.1 Introduction:

In the early 1800's mechanical puppets were first built in Europe, just for entertainment value. And these were called robots since their parts were driven by linkage and cams and controlled by rotating drum selectors. In 1801 Joseph Maria Jacquard made the next great change and invented the automatic draw loom. The draw looms would punch cards and was used to control the lifting of thread in fabric factories. This was the first to be able to store a program and control a machine. After that there were many small changes in robotics. The first industrial robots were Unimates developed by George Devol and Joe Engelberger in the late 50's and early 60's. The first patent was by Devol but Engelberger formed Unimation which was the first market robots. So Engelberger has been called the "father of robotics". For a while the economic viability of these robots proved disastrous and thing slowed down for robotics. But the industry recovered and by the mid-80's robotics was back on track.[1]

George DevolJr, in 1954 developed the multi jointed artificial arms which lead to the modern robots. But mechanical engineer Victor Scheinman developed the truly flexible arm known as the Programmable Universal Manipulation Arm (PUMA) [1]. In 1950 Isaac Asimov came up with laws for robots and these were:

i. A robot may not injure a human being, or through inaction allow a human being to come to harm.

ii. A robot must obey the orders given it by human beings, except where such orders would conflict with the first law

iii. A robot must protect its own existence as long as such protection does not conflict with the first or second law.[2]

Mobile Robotics moved into its own in 1983 when Odetics introduced this six-legged vehicle which was capable of climbing over objects. This robot could lift over 5.6 times its own weight parked and 2.3 times it weight moving. [3] In 2000 Sony unveils humanoid robots, the Sony Dream Robots (SDR) at Robodex. SDR is able to recognize 10 different faces, expresses emotion through speech and body language, and can walk on flat as well as irregular surfaces. In 2005 the Korean Institute of Science and Technology (KIST), creates HUBO, and claims it is the smartest robot in the world. This robot is linked to a computer via a high speed wireless connection; the computer does all of the thinking for the robot.[1]

1.2 Application area:

Line followers can be used to deliver mail within an office building and deliver medications in a hospital. The technology has been suggested for running buses and other mass transit systems, and may end up as part of autonomous cars navigating the freeway. The line follower can be used in guidance system for industrial robots moving on shop floor. An example might be in a warehouse where the robots follow 'tracks' to and from the shelves they stock and retrieve from. A line follower robot can be used in military as spy kids or in many other applications.[4]

2.1 Introduction:

Robotics has achieved its greatest success to date in the world of industrial manufacturing. Robot arms, or *manipulators*, comprise a 2 billion dollar industry. Bolted at its shoulder to a specific position in the assembly line, the robot arm can move with great speed and accuracy to perform repetitive tasks such as spot welding and painting (figure 2.1). In the electronics industry, manipulators place surface-mounted components with superhuman precision, making the portable telephone and laptop computer possible. Yet, for all of their successes, these commercial robots suffer from a fundamental disadvantage: lack of mobility. A fixed manipulator has a limited range of motion that depends Picture of auto assembly plant-spot welding robot of KUKA and a parallel robot Delta of SIGDemaurexSA (invented at EPFL [140]) during packaging of chocolates.on where it is bolted down. In contrast, a mobile robot would be able to travel through outthe manufacturing plant, flexibly applying its talents wherever it is most effective. This book focuses on the technology of mobility: how can a mobile robot move unsupervised through real-world environments to fulfill its tasks? The first challenge is locomotion itself. How should a mobile robot move, and what is it about a particular locomotion mechanism that makes it superior to alternative locomotion mechanisms? Hostile environments such as Mars trigger even more unusual locomotion mechanisms (figure 2.2). In dangerous and inhospitable environments, even on Earth, such *teleoperated*systems have gained popularity. In these cases, the low-level complexities of the robot often make it impossible for a human operator to directly control its motions. The human performs localization and cognition activities, but relies on the robot's control scheme to provide motion control. For

example, Plustech's walking robot provides automatic leg coordination while the human operator chooses an overall direction of travel (figure 2.1).depicts an underwater vehicle that controls six propellers to autonomously stabilize the robot submarine in spite of underwater turbulence and water currents while the operator chooses position goals for the submarine to achieve.

Other commercial robots operate not where humans *cannot* go but rather share space with humans in human environments. These robots are compelling not for reasons of mobility but because of their *autonomy*, and so their ability to maintain a sense of position and to navigate without human intervention is paramount [5]

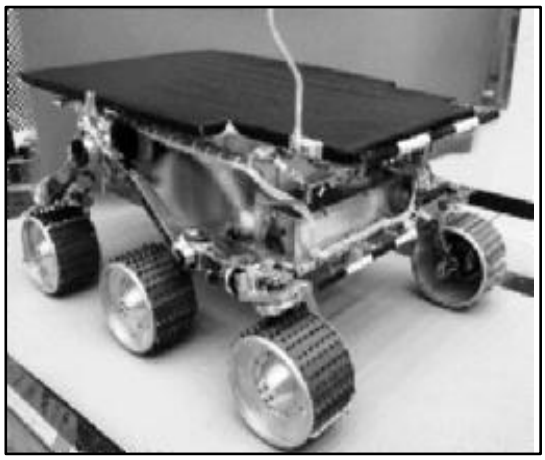


Figure (2.1): legged robot

The mobile robot Sojourner was used during the Pathfinder mission to explore Mars in summer 1997. It was almost completely teleoperated from Earth. However, some on-board sensors allowed for obstacle detection.(http://ranier.oact.hq.nasa.gov/telerobotics_page/telerobotics.shm). © NASA/JPL[5]

Mobile robotics is a young field. Its roots include many engineering and science disciplines,from mechanical, electrical and electronics engineering to computer, cognitive and social sciences. Each of these parent fields has its share of introductory textbooks that excite and inform prospective students, preparing them for future advanced coursework and research. Our objective in writing this textbook is to provide mobile robotics with such a preparatory guide.This book presents an introduction to the fundamentals of mobile robotics, spanning the mechanical, motor, sensory, perceptual and cognitive layers that comprise our field of study. A collection of workshop proceedings and journal publications could present the new student with a snapshot of the state of the art in all aspects of mobile robotics. But here we aim to present a foundation a formal introduction to the field. The formalism and analysis herein will prove useful even as the frontier of the state of the art advances due to the rapid progress in all of mobile robotics' sub-disciplines.We hope that this book will empower both the undergraduate and graduate robotics student with the background knowledge and analytical tools they will need to evaluate andThe origins of the this book bridge the Atlantic Ocean. Theauthors have taught courses on Mobile Robotics at the undergraduate and graduate level at Stanford University, ETH Zurich, Carnegie Mellon University and EPFL (Lausanne). Their combined set of curriculum details and lecture notes formed the earliest versions of this text. We have combined our individual notes, provided overall structure and then test-taught using this textbook for two additional years before settling on the current, published text. For an overview of the organization of the book and summaries of individual chapters.[6]

Finally, for the teacher and the student: we hope that this textbook proves to be a fruitful launching point for many careers in mobile robotics. That

would be the ultimate reward.even critique mobile robot proposals and artifacts throughout their career. This textbook is suitable as a whole for introductory mobile robotics coursework at both the undergraduate and graduate level. Individual chapters such as those on Perception or Kinematics can be useful as overviews in more focused courses on specific sub-fields of robotics.[6]

2.2: Legged Mobile Robots:

Legged locomotion is characterized by a series of point contacts between the robot and the ground. The key advantages include adaptability and maneuverability in rough terrain. Because only a set of point contacts is required, the quality of the ground between those points does not matter so long as the robot can maintain adequate ground clearance. In addition, a walking robot is capable of crossing a hole or chasm so long as its reach exceeds the width of the hole. A final advantage of legged locomotion is the potential to manipulate objects in the environment with great skill. An excellent insect example, the dung beetle, is capable of rolling a ball while locomoting by way of its dexterous front legs. The main disadvantages of legged locomotion include power and mechanical complexity. The leg, which may include several degrees of freedom, must be capable of sustaining part of the robot's total weight, and in many robots must be capable of lifting and lowering the robot. Additionally, high maneuverability will only be achieved if the legs have a sufficient number of degrees of freedom to impart forces in a number of different directions.[6]

2.3: Analysis of Line Sensor Configuration for the Advanced Line Follower Robot:

A line follower robot is basically a robot designed to follow a ‘line’ or path already predetermined by the user. This line or path may be as simple as a physical white line on the floor or as complex path marking schemes e.g. embedded lines, magnetic markers and laser guide markers. In order to detect these specific markers or ‘lines’, various sensing schemes can be employed [1]. These schemes may vary from simple low cost line sensing circuit to expansive vision systems. The choice of these schemes would be dependent upon the sensing accuracy and flexibility required. From the industrial point of view, line following robot has been implemented in semi to fully autonomous plants. In this environment, these robots functions as materials carrier to deliver products from one manufacturing point to another where rail, conveyor and gantry solutions are not possible[1]. Apart from line following capabilities, these robots should also have the capability to navigate junctions and decide on which junction to turn and which junction ignore. This would require the robot to have 90 degree turn and also junction counting capabilities. To add on to the complexity of the problem, sensor positioning also plays a role in optimizing the robots performance for the tasks mentioned earlier[2]. This paper attempts to present a simple set of experiments on sensor positioning and also controlling strategy to enable junction counting and also 90 degree turn accuracy. These strategies are tested on a basic test robot base (refer to figure 1) system on a test pitch based upon ROBOCON 2006 [7]requirement as shown in figure(2.2)

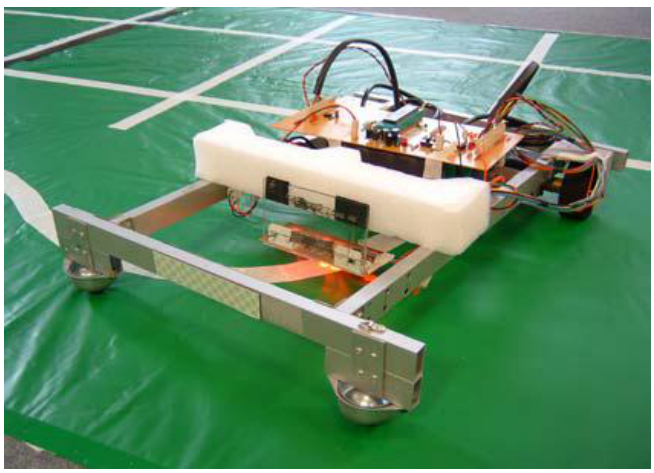


Figure (2.2): Advanced Line Follower (ALF) Robot

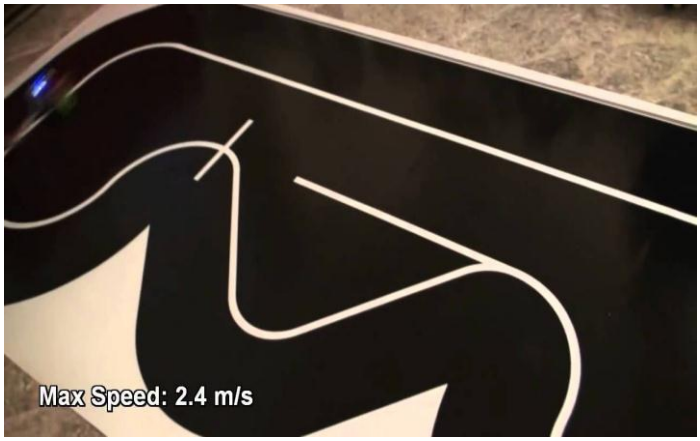


Figure (2.3): ROBOCON 2006 pitch requirement

2.4: Application:

• The robots have potential application in areas where a vehicle or a mechanic automatic system may exist

- Areas of application:

1. Support to medical services – SERVICE ROBOTS
a. Transportation of food, medication, medical exams
b. Automation of pharmacy service
2. Automatic cleaning of (large) area
a. Supermarkets, airports, industrial sites
b. Glass cleaning
c. Domestic vacuum-cleaner
3. Client support
a. Museum tours, exhibitions guides
4. Agricultural
a. Fruit and vegetable picking, fertilization, planting
5. Forests

a. Cleaning, fire preventing, tree cutting

b. Areas of application (ctn):

6. Hazard Environments

a. Inspection of hazard environments (catastrofic areas, vulcanos, nuclear

7. power plants, oil tanqs)

a. Inspection of gas or oil pipes, and power transmis

b. Oil tank cleaning

8. Construction and demolishing

9. Space

a. Space exploration

b. Remote inspection of space stations

10. Military

a. Surveilance vehicles

b. Monitoring vehicles

11. Forests

a. Cleaning, fire preventing, tree cutting

b. Areas of application (ctn):

12. Material Handling

a. AGVs, SGVs, LGVs

13. Safety

a. Surveillance of large areas, buildings, airports, car parking lots

14. Civil Transportation

a. Inspection of airplanes, trains,

15. Elderly and Handicapped

a. Assistance to handicapped or elderly people, helping in tansportation, health care,

16. Entertainment

a. RobotDog

b. Aibo – Robot dog from Sony [7]

3.1 Introduction:

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source,which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time.For advanced Arduino users, prowl the web; there are lots of resources.The Arduino project was started in Italy to develop low cost hardware for interaction design. An overview is on the Wikipedia entry for Arduino. The Arduino home page is http://www.arduino.cc/.The Arduino hardware comes in several flavors. In the United States, Sparkfun (www.sparkfun.com) is a good source for Arduino hardware.This guide covers the Arduino Uno board (Sparkfun DEV-09950, $29.95), a good choice for students and educators. With the Arduino board, you can write programs and create interface

circuits to read switches and other sensors, and to control motors and lights with very little effort. Many of the pictures and drawings in this guide were taken from the documentation on the Arduino site, the place to turn if you need more information. The Arduino section on the ME 2011 web site, https://sites.google.com/a/umn.edu/me2011/, covers more on interfacing the Arduino to the real world..[8] As shown in Fig(3.1): below

Fig (3.1): Ardunio UNO

The Arduino programming language is a simplified version of C/C++. If you know C,programming the Arduino will be familiar. If you do not know C, no need to worry as only a few commands are needed to perform useful functions. An important feature of the Arduino is that you can create a control program on the host PC,download it to the Arduino and it will run automatically. Remove the USB cable connection to the PC, and the program will still run from the top each time you push the reset button. Remove the battery and put the Arduino board in a closet for six months. When you reconnect the battery, the last program you stored will run. This means that you connect the board to the host PC to develop and debug your program, but once that is done, you no longer need the PC to run the program.[8]

Arduino is an open source physical computing platform based on a simple input/output (I/O) board and a development environment that implements the Processing language (www.processing.org). Arduino can be used to develop standalone interactive objects or can be connected to software on your computer (such as Flash, Processing, or Max/MSP). The boards can be assembled by hand or purchased preassembled; the open source IDE (Integrated Development Environment) can be downloaded for free from www.arduino.cc Arduino is different from other platforms on the market because of these

features:

1. It is a multiplatform environment; it can run on Windows, Macintosh,and Linux
2. It is based on the Processing programming IDE, an easy-to-use development environment used by artists and designers
3. You program it via a USB cable, not a serial port. This feature is useful,because many modern computers don't have serial ports.
4. It is open source hardware and software—if you wish, you can download the circuit diagram, buy all the components, and make your own, without paying anything to the makers of Arduino
5. The hardware is cheap. The USB board costs about €20 (currently, A bout US$35) and replacing a burnt-out chip on the board is easy and costs no more than €5 or US$4. So you can afford to make mistakes.
6. There is an active community of users, so there are plenty of people who can help you.
7. The Arduino Project was developed in an educational environment and is therefore great for newcomers to get things working quickly.

3.2 What Is Physical Computing?

Physical Computing uses electronics to prototype new materials for designers and artists. It involves the design of interactive objects that can communicate with humans using sensors and actuators controlled by a behaviour implemented as software running inside a microcontroller (a small computer on a single chip). In the past, using electronics meant having to deal with engineers all the time, and building circuits one small component at the time; these issues kept creative people from playing around with the medium directly. Most of the tools were meant for engineers and required extensive knowledge. In recent years, microcontrollers have become cheaper and easier to use, allowing the creation of better tools.The progress that we have made with Arduino is to

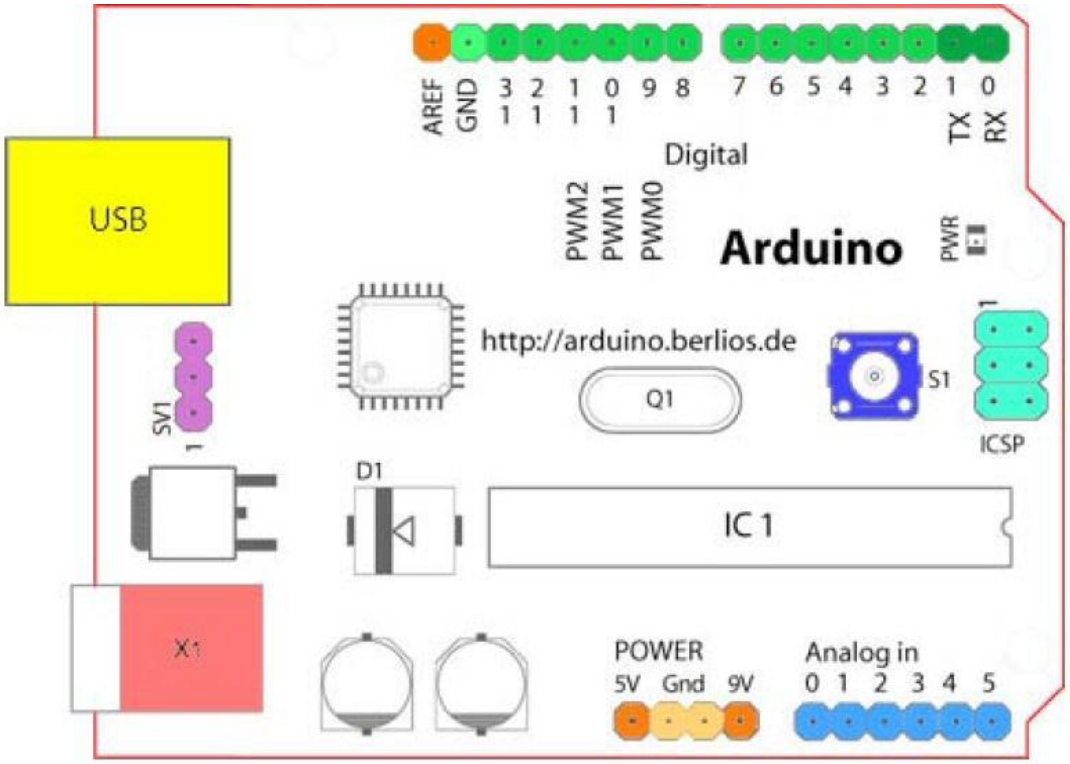
bring these tools one step closer to the novice, allowing people to start building stuff after only two or three days of a workshop. With Arduino, a designer or artist can get to know the basics of electronics and sensors very quickly and can start building prototypes with very little investment.

The Arduino philosophy is based on making designs rather than talking about them. It is a constant search for faster and more powerful ways to build better prototypes. We have explored many prototyping techniques and developed ways of thinking with our hands. Classic engineering relies on a strict process for getting from A to B; the Arduino Way delights in the possibility of getting lost on the way and finding C instead.This is the tinkering process that we are so fond of—playing with the medium in an open-ended way and finding the unexpected. In this search for ways to build better prototypes, we also selected a number of software packages that enable the process of constant manipulation of the software and hardware medium. The next few sections present some philosophies, events, and pioneers that have inspired the Arduino Way.

3.3 Arduino UNO:

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, aUSB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The

Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards. The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.[9]



The pins of the arduino UNO are shown in fig (3.2)

Power pins are as follows:

1. Th VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
2. 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
3. 3V3. A 3.3 volt supply generated by the on-board regulator Maximum current draw is 50 mA.
4. GND. Ground pins.

3.3.1 Input And Output

Each of the 14 digital pins on the Uno can be used as input or output, using pinMode(), digitalWrite(), anddigitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions: Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. TThese pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip. External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details. PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is

not currently included in the Arduino language. LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off [9]

3.3.2 Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required.. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-toserial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.[9]

3.3.3 Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials. The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to

it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details. The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).[9]

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration

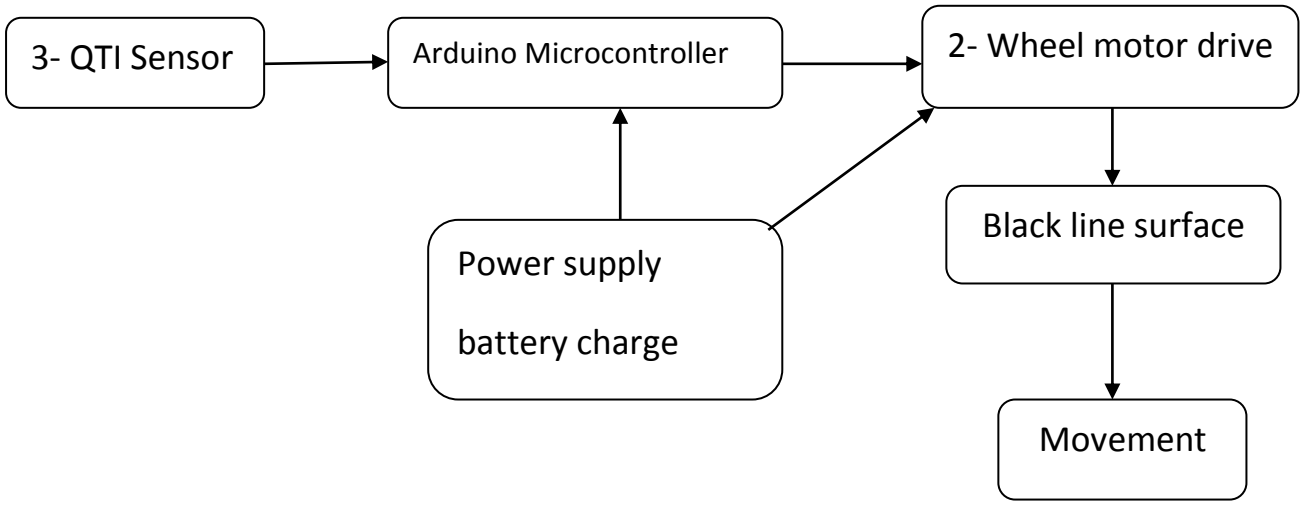
or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.[9]

3.3.4 USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.[6] The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multipleof the 100 mil spacing of the other pins.[9]

4.1 Introduction:

In this project the two wheel forward motor used to determine the direction with one at the back is used to assign only the following components were used in the project we used the microcontroller ardunio device and the use of three sensor from type QTI sensor and the number of QTI are used is 3 first of the QTI sensor connector to the left and the second of the QTI sensor connector to the center and the third connector to the right and also used dc motor drives .AS to explain at this paper and shown the block diagram of line follower mobile robot:

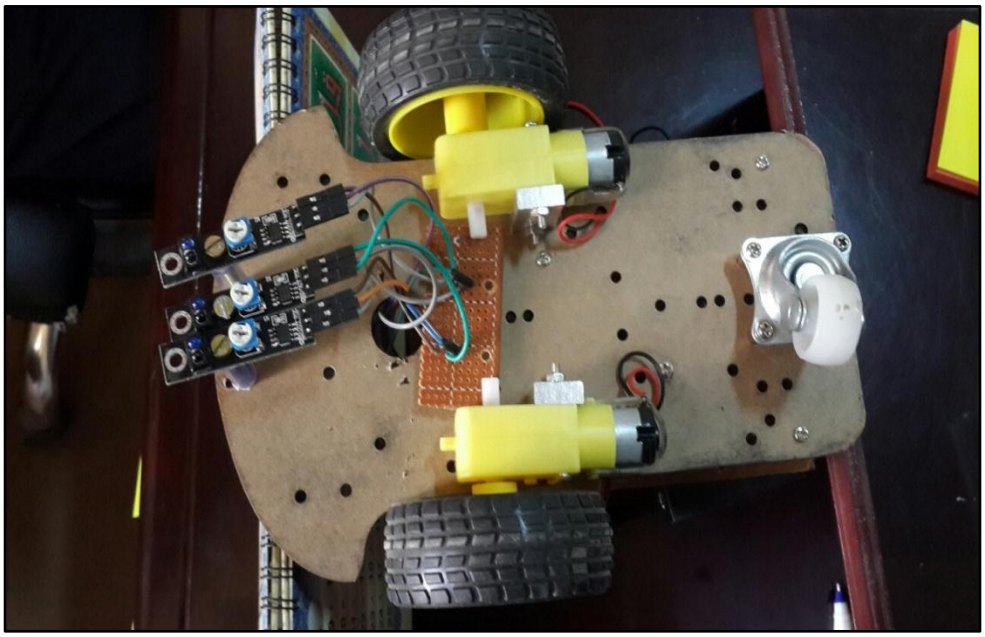


Fig(4.1):Block diagram of line follower

4.2 Motors & Wheels:

4.2.1 Driven Wheels:

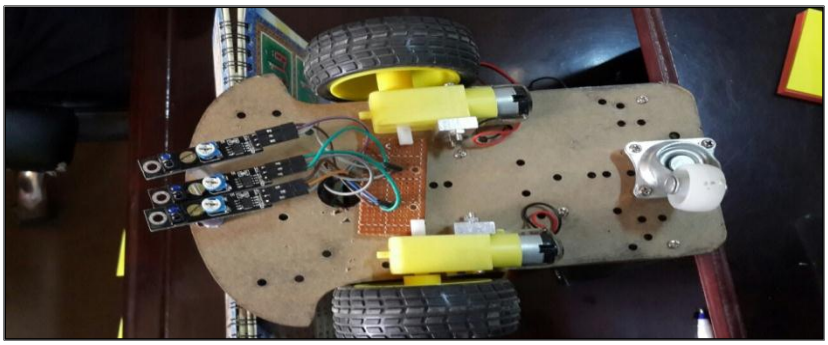
Weight-bearing wheels not driven by motors, directly or indirectly, are considered driven wheels. Driven wheels unable to swivel left and right can cause excess friction when the robot turns. Caster wheels are a special kind of driven wheel that allows the wheels to swivel left and right, eliminating unnecessary friction when the robot turns.[10] As shown in fig below:



Fig(4.2): Driven wheel

4.2.3 Omni-Directional Wheels:

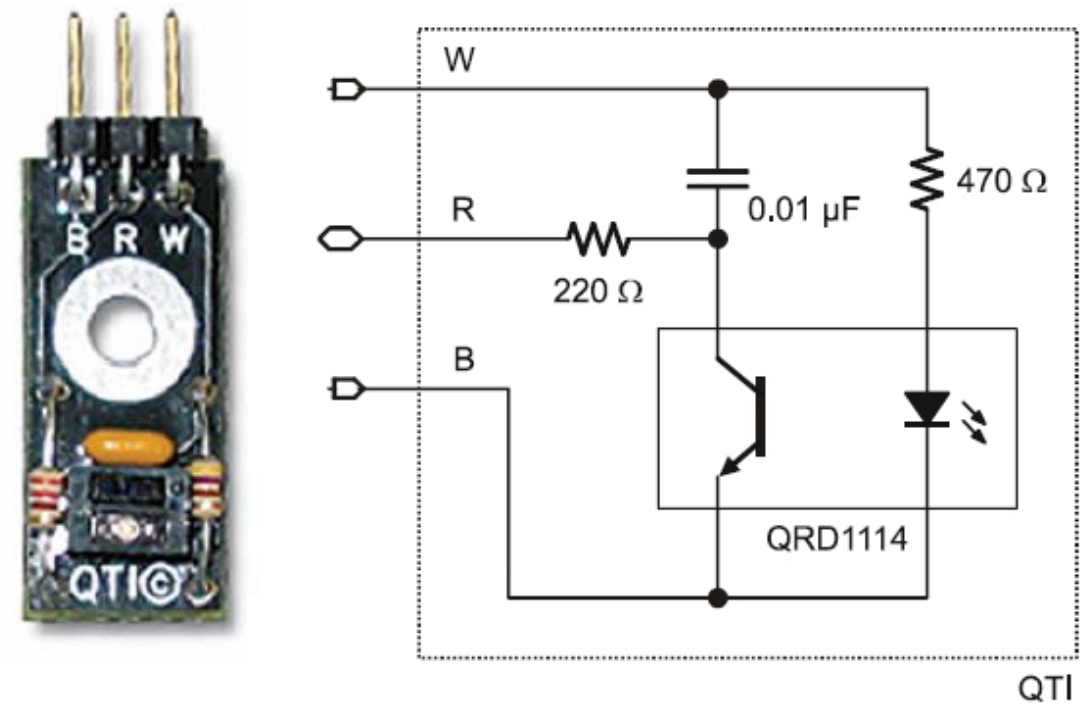
The TETRIX Omni-directional wheels, or Omni-wheels, are comprised of one large wheel with many smaller wheels along its circumference. The smaller wheels are perpendicular to the main wheel, allowing it to roll both forward and backward, and side-to-side, naturally.[10]

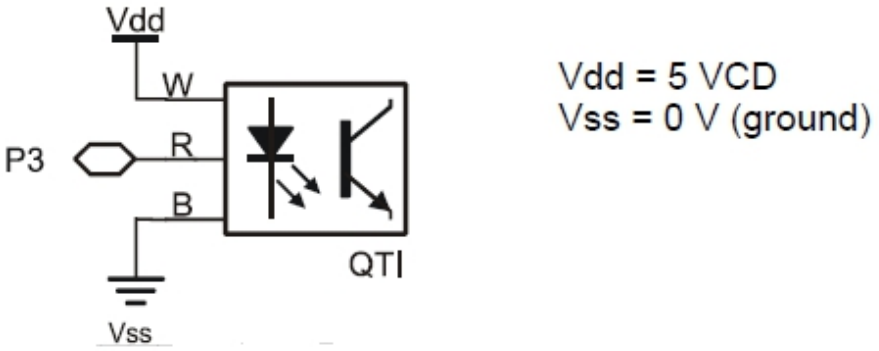


Fig(4.3):Wheel backing

4.3 QTI Sensor :

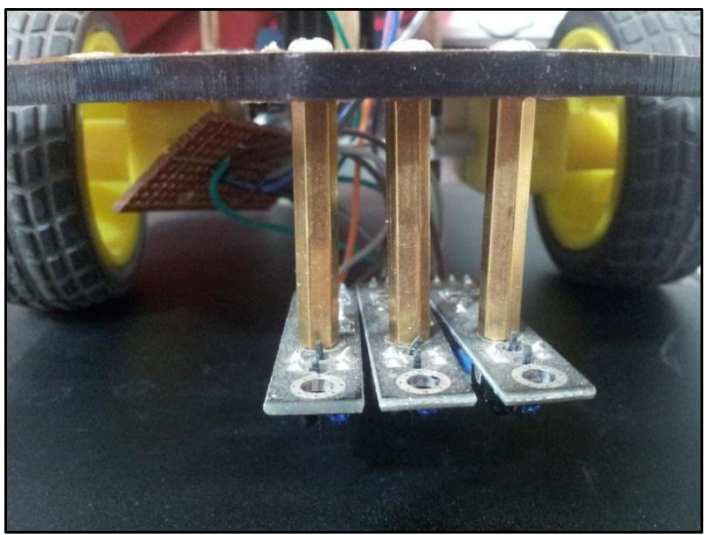
The QTI sensor is a close-proximity infrared emitter and receiver pair mounted on a tiny PCB. It can be used as an analog sensor to differentiate between different levels of infrared reflectivity. It can also be used as a purely digital device that returns a 1 when it detects a black line or a 0 if it detects a white background. An array of four QTI sensors used as digital devices makes an effective and flexible line-follower for your small robot.**[11]**





Fig(4.4): QTI sensor

It was used in this project 3 QTI Sensor uses a first of sensor for sensing motor wheel drive on the right and the second QTI is used to sensing motor wheel drive on the left and the third QTI Sensor used to sensing for the middle. As shown in figure (4:5) below:

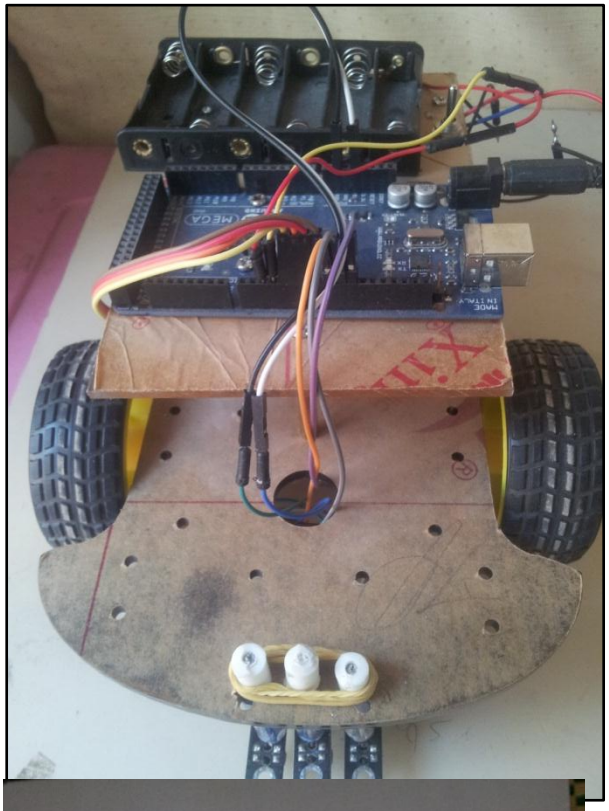


Fig(4.5): QTI Sensor

4.4 Arduino UNO:

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, aUSB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduno, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards. The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage

the board. The recommended range is 7 to 12 volts.[9] As shown in fig.(4.6).



Fig(4.6): Ardunio Microcontroller

There are two line test the robot to follow there as shown in fig.(4:7).

Fig.(4.7): Two Line track

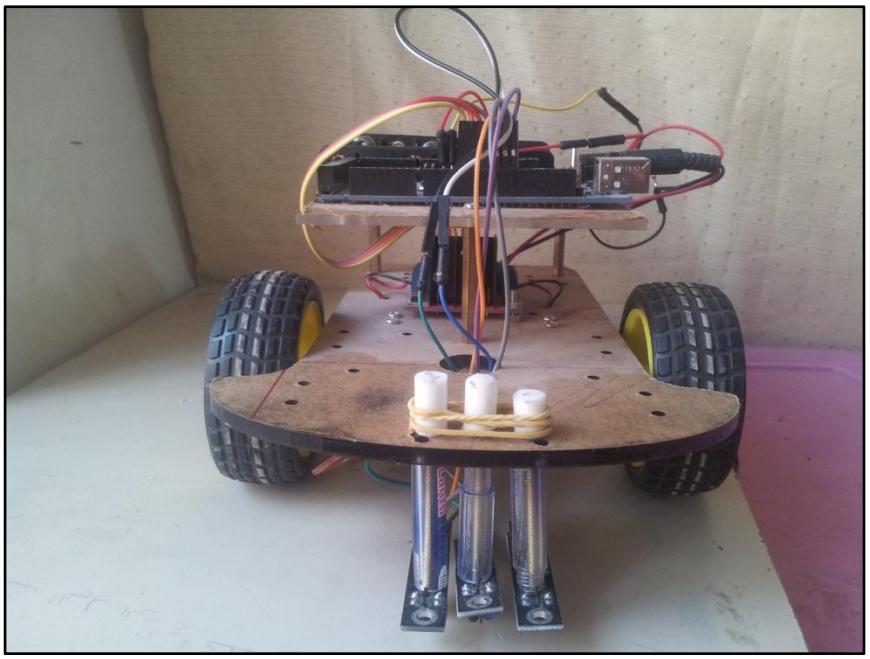
4.5 Software

The line used in the project is Black color and used tipp for the path Selection and it was select many paths to be tested robot.

4.7 movement:

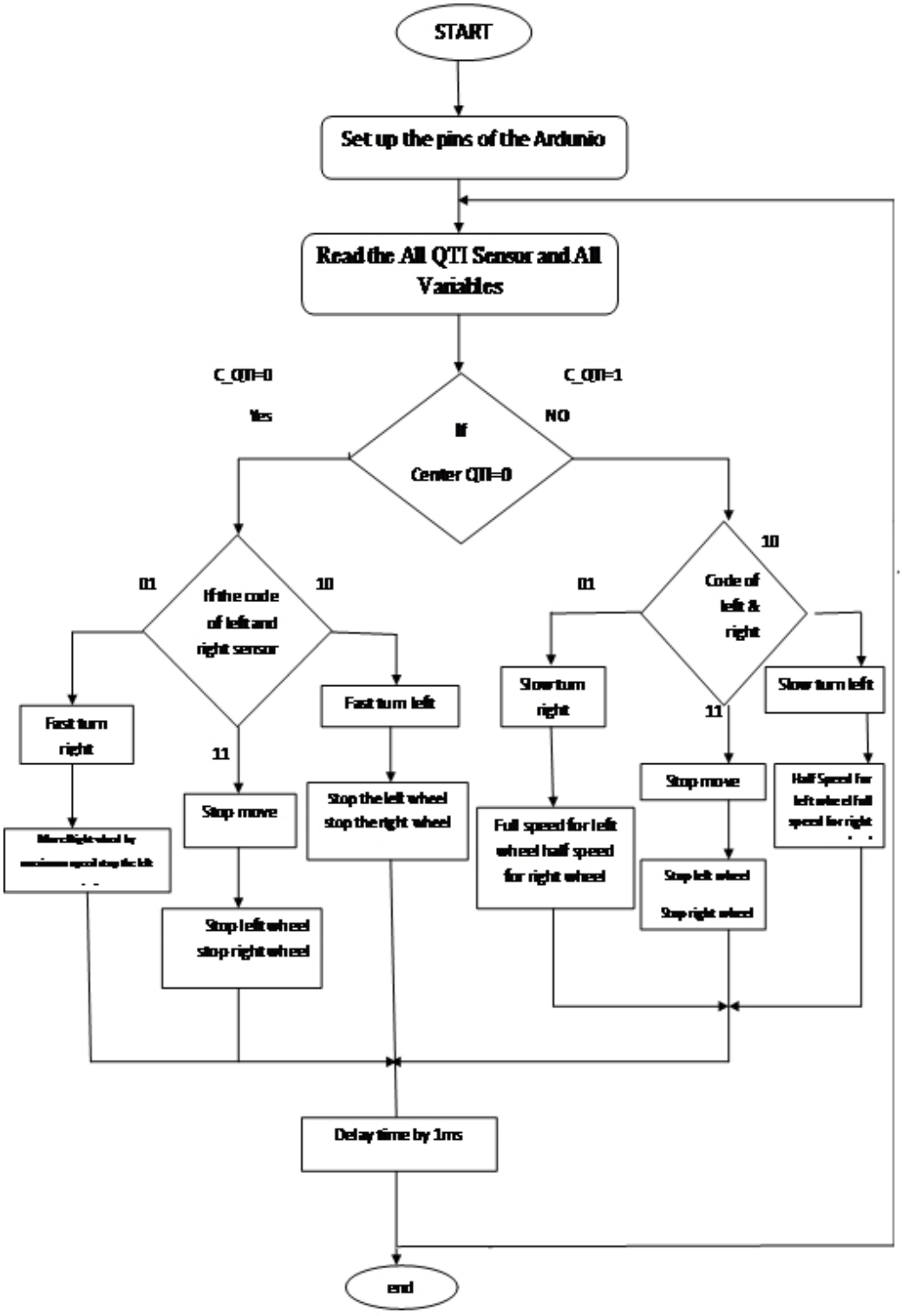
Motion shall be according to the path that determines.

And that the figure of this project:



Fig(4.8): line follower mobile robot

The microcontroller is read the output of the QTI sensoers then make the dicession of the movement forward, turn left or right fast or slow or stop when all the sensor detect black serface (output signal of QTI sensoris 1). The flowchart of the program is shown in fog.(4.9):



Fig(4.9): flow chart of the line follower mobile robot

And the code is:

```
//Line follower_Robort
int RS=8;
int CS=9;
int LS =10;
int spd _LM=3;
int spd_RM=2;

int LM_A=4;
int LM_B=5;
int RM_A=6;
int RM_B=7;
int spd_100;
int backmove_index=0;

int rv,cv,lv;
voidsetup()
{
pinMode(RS,INPUT);
pinMode(CS,INPUT);
pinMode(LS,INPUT);

PINMode(spd_LM,OUTPUT);
PINMode(spd_RM,OUTPUT);
pinMode(LM_A,OUTPUT);
pinMode(LM_B,OUTPUT);

pinMode(RM_A,OUTPUT);
```

```
pinMode(RM_B,OUTPUT);

digitalWrite(LM_A,HIGH);
digitalWrite(LM_B,LOW);

digitalWrite(RM_A,HIGH);
digitalWrite(RM_B,LOW);

analogWrite(spd_LM,0);

analogWrite(spd_RM,0);

delay(1000);
serial.begin(9600);
}
Void loop()
{
rv=digitalRead(RS);
cv=digitalRead(CS);
lv=digitalRead(LS);
Searial.print(rv);
Searial.print('\t');
Searial.print(cv);
Searial.print('\t');
Searialprintin(iv);

// //delay(100);
If(backmove_index==1)
{if(rv==1 || cv==1 || lv==1)
```

```
{correct_move();}
}
If(cv==1)// center at black line
{
If(rv==0 && lv==0)    forward();
else if(rv==1 && lv==0)  slow_left();
else if(rv==0 && lv==1)  slow_right();
else if(rv==1 && lv==1)  stop_move ();
}
else //if(cv==0)center at waith back graound
}
If(rv==1 && lv==1)   stop_move();
else If(rv==1 && lv==0)   fast_left();
else If(rv==0 && lv==1)   fast_right();
// else If(rv==0 && lv==0)   back_move();//search();
}
//delay(10);
}
////////////
Void forward()
{
analogWrite(spd_LM,spd);
analogWrite(spd_RM,spd);
}
Void fast_left()
{
AnalogWrite(spd_LM,0);
AnalogWrite(spd_RM,spd);
}
```

```
void slow_left()
{
analogWrite(spd_LM,spd/2);
analogWrite(spd_RM,spd);
}
void fast_right()
{
analogWrite(spd_LM,spd);
analogWrite(spd_RM,0);
}
void slow_right()
{
 analogWrite(spd_LM,spd);
analogWrite(spd_RM,spd/2);
}
void stop_move()
}
analogWrite(spd_LM,0);
analogWrite(spd_RM,0);
}
Voidback_move(void)
{
// if (backmove_index==1)
//{
Stop_move();
digitalWrite(LM_A;LOW);
digitalWrite(LM_B;HIGH);

digitalWrite(RM_A;LOW);
```

```
digitalWrite(RM_B;HIGH);
forward();
backmove_index=1;
//}

}
void correct-move(void)
{
Stop_move();
digitalWrite(LM_A;HIGH);
digitalWrite(LM_B;LOW);

digitalWrite(RM_A;HIGH);
digitalWrite(RM_A;LOW);
backmove_index=0;
}
```

5.1 Conclusion:

The Line follower robot works successfully to track on the black line. Above the white surface (art paper) there are some black lines in different directions. The robot still good enough to sense the line and follows the track**.**

5.2 Future work:

1- Software control of the line type (dark or light) to make automatic detection possible.
2- “Obstacle detecting sensors” to avoid physical obstacles and continue on the line.
3- Distance sensing and position logging & transmission